Journal of Rare Cardiovascular Diseases
ISSN: 2299-3711 (Print) | e-ISSN: 2300-5505 (Online)
www.jrcd.eu
RESEARCH ARTICLE

A COMPARATIVE EMPIRICAL EVALUATION OF QUANTUM MACHINE LEARNING MODELS FOR BREAST CANCER DETECTION

Levadala Bhavya[1], Dr. G. Murali[2]
[1] Assistant Professor (A), Dept. Of CSE, JNTUA College of Engineering, Pulivendula
[2]Assistant Professor, Dept. Of CSE, JNTUA College of Engineering, Pulivendula

*Corresponding Author

Article History
Received: 09.09.2025
Revised: 07.10.2025
Accepted: 31.10.2025
Published: 29.11.2025

Abstract: Quantum Machine Learning (QML) is gaining significant traction across various fields for its potential to vastly outperform traditional computational methods, thanks to the advanced computational capabilities of quantum computers. This potential is particularly promising in healthcare, where early and accurate diagnosis of the most common cancer in the world, breast cancer, among others, can greatly enhance treatment outcomes. In light of this, our research explores the application of QML algorithms, specifically Quantum Support Vector Machine (QSVM), Variational Quantum Classifier (VQC), Quantum Random Forest (QRF), and Quantum k-Nearest Neighbors (QkNN), for breast cancer diagnosis. Utilizing the IBM Qiskit framework, we apply these QML algorithms to classify feature attributes derived from Fine Needle Aspirate (FNA) images of breast cancer tissues. These attributes are converted into qubit structures through Principal Component Analysis (PCA), enabling the quantum models to process the high-dimensional biomedical data. Classical machine learning methods such as Support Vector Classifier (SVC), Random Forest (RF), and K-Nearest Neighbors (KNN) are tested against these QML models on the same dataset.
Our findings highlight that QML models, particularly QRF and QSVM, exhibit performance levels that closely rival their classical counterparts, with QRF achieving the highest training (0.91) and testing (0.90) accuracies among the quantum algorithms studied. Despite the promising capabilities of QML models to effectively handle complex biomedical data, classical ML models currently show higher accuracy and reliability. To fully harness the power of quantum algorithms for medical diagnostics, this disparity highlights the need for ongoing improvements in quantum computing hardware and optimization of algorithms. The purpose of this study is to compare and contrast conventional and quantum machine learning methods for early breast cancer diagnosis in the hopes of drawing useful conclusions. By highlighting the commendable performance of QML models and identifying areas for further development, We add to the increasing amount of evidence that supports using quantum computing for medical diagnostics with our work. The outcomes of this research suggest a future where quantum algorithms could match or even surpass classical methods, increasing the precision of breast cancer detection and therapy by a substantial margin.

Keywords: Quantum Machine Learning (QML), Breast Cancer Diagnosis, Principal Component Analysis (PCA), Quantum Support Vector Machine (QSVM), Variational Quantum Classifier (VQC), Quantum Random Forest (QRF), Quantum k-Nearest Neighbors (QkNN), Accuracy Measures

INTRODUCTION

With the rise of quantum computing, a fresh approach to artificial intelligence is taking shape. Given the immense processing capacity of quantum computers, quantum machine learning might potentially surpass traditional machine learning when it comes to computation speed. Evidence for this comes from research showing that quantum learning algorithms outperform their classical counterparts when it comes to data classification [1]. The potential for quantum machine learning to provide more accurate results is a further advantage, as shown in [2]. However, there is still a need for progress in the development of technologies that allow intelligent data processing to take advantage of this new paradigm in computing.

Breast cancer, the most prevalent cancer among women globally, has seen improved survival rates thanks to advances in detection and treatment, with a 5-year survival rate exceeding 90% for early-stage diagnoses in many regions. Key risk factors include gender, age, genetics, and lifestyle, while disparities in incidence and outcomes highlight the need for ongoing research and equitable healthcare access [3]. Enhanced screening and targeted therapies are crucial in managing this widespread disease effectively. The breast cancer diagnosis rate for women is still second only to that for men. Cancer of the breast often starts in the breast's cells, specifically in the breast's ducts or lobules. At the first sign of infection, a patient may perceive redness, swelling, skin irritation, a breast lump, nipple retractions, lymph node alterations, retraction, etc. Trustworthy technology [4] that can successfully detect and cure this kind of cancer has been steadily improving over the previous many decades. The exponential rise of available data and the development of increasingly powerful computing hardware have allowed computer science researchers to properly categorize real-world data using a range of data mining [5] and machine learning [6] methodologies.

Breast cancer diagnosis is still mainly a binary classification issue, with two possible types of tumors: malignant (cancerous) and benign (non-cancerous) [6]. The majority of breast cancer datasets may be modeled by several of the current machine learning classifiers, including k-nearest neighbor, supported vector machines (SVMs), decision trees, random forests, and artificial neural networks. While most of the classifiers listed above need high-end computational capacity for classifier training, this becomes more problematic as the issue size and number of determining variables (i.e., features) grow.

Exploring these new areas and using machine learning techniques to do the necessary computations with less processing power is a natural progression given the developments in other types of computing. Using current noisy intermediate-scale quantum technology and quantum computing principles, it is now feasible to speed up conventional machine learning classifiers without sacrificing classification accuracy. Methods utilized in quantum machine learning, which are based on the concepts of quantum speedup, have the potential to surpass conventional methods [7]. In addition, it can revolutionize machine learning, bringing it closer to creating the best categorization systems possible. For problems like breast cancer classification, which takes a huge number of characteristics into account to complete the classification assignment, SVM classifiers—one of the most developed conventional classifiers—are the ideal option. Such difficult classification tasks are likely to be well-suited for nonlinear support vector machines (SVMs), which, via the use of their kernel technique [8], may translate input into higher dimensional space. Due to its computationally costly kernel matrix construction and solution, It is challenging to compute the nonlinear SVM classification method using a conventional computer system, especially when dealing with the least-squares formulation or quadratic dual issue [9]. Making use of quantum support vector machines [11] and quantum machine learning [10]. may provide a substantial improvement in the computational capacity of the traditional SVM classifier. Since SVMs reduce optimization problems to sets of linear equations, quantum SVM can gain computing speedups via two means: first, by calculating the kernel matrix more quickly; and second, by effectively solving these linear equations on quantum hardware. There are exciting new possibilities for improving classification problems like breast cancer classification using quantum machine learning algorithms like Quantum k-Nearest Neighbors (QkNN) [12], Quantum Random Forests (QRF), and Variational Quantum Circuits (VQC) [13]. To effectively handle high-dimensional data, VQC integrates classical and quantum computing components, making use of quantum circuits. By applying quantum concepts to classical Random Forests, QRF allows for scalability and quicker processing of decision trees. QkNN is a quantum-based version of the well-known k-Nearest Neighbors method that allows for fast neighbor search in feature spaces with many dimensions. These quantum methods take advantage of superposition and parallelism, two quantum properties, to potentially outperform classical classifiers like Support Vector Machines (SVMs) in terms of computational speed and accuracy, particularly when dealing with complicated classification problems involving big feature sets.

In the proposed work, we meticulously implemented and compared Quantum Machine Learning (QML) models against classical Machine Learning (ML) approaches for breast cancer prediction, leveraging the IBM Qiskit Aqua library [14] within a simulated environment on a classical computer. The procedure commenced with dataset preparation, focusing on cell nuclei characteristics from Fine Needle Aspirate (FNA) images, followed by feature reduction using PCA [15] to streamline the complexity for both quantum and classical algorithms. The quantum models— By encoding the processed characteristics into qubits, the following quantum classifiers were simulated: Quantal k-Nearest Neighbors (QkNN) [12], Quantum Random Forest (QRF) [13], Quantum Support Vector Machine (QSVM) [11], and Variational Quantum Classifier (VQC) [13]. At the same time, we used several standard models such as Support Vector Classifier (SVC), Random Forest (RF), and K-Nearest Neighbors (KNN). We trained and tested these models thoroughly, using metrics like F1-score, recall, accuracy, and precision [13]. This comprehensive implementation and evaluation process underscored the promising potential and current limitations of QML in medical diagnostics, highlighting the need for ongoing optimization and development to harness the full capabilities of quantum computing in healthcare analytics.

The main contribution of this research is

- To develop a better machine-learning breast cancer classifier, we used the Qiskit package to build and simulate quantum systems using QML classification algorithms.
- To optimize the data for quantum computing, we used the Principal Component Analysis (PCA) technique to minimize the amount of features until it was in line with the number of qubits present.
- Developed a quantum-powered classification model that solves breast cancer detection in logarithmic order, unlike existing polynomial methods. This innovation may speed categorization and reduce computing power.
- Used criteria including accuracy, precision, recall, and F1-score to compare QML algorithm performance to that of conventional machine learning algorithms.
- Measured the computational cost of solving the breast cancer detection issue, comparing the speed and resource consumption of quantum and conventional techniques.

The implementation of QML models, particularly QRF and QSVM, exhibit performance levels that closely

rival their classical counterparts, with QRF achieving the highest training (0.91) and testing (0.90) accuracies among the quantum algorithms studied. Despite the promising capabilities of QML models to effectively handle complex biomedical data, classical ML models currently show higher accuracy and reliability.

What follows is an outline of the rest of the paper: Following a short introduction in Section 2, Section 3 delves into the literature study and preliminary results. In Section 4, we go over the proposed approach. Section 5 presents the findings and discussion, followed by the conclusion and future implications.

2 BASIC PRELIMINARIES AND RELATED WORKS

Within the realm of physical science, The basic discontinuous unit of a "physical quantity" is a "quantum." Quantum theory studies the likelihood of bidirectional (wave-particle) particles like quantum particles in a particular spatial place [16]. As an offshoot of Machine Learning (ML), which processes and analyzes data using different decision-making models, Over the last several decades, Quantum Machine Learning (QML) has seen significant growth and transformation within the realm of computer science. The need for effective data management is growing as the amount of data continues to rise at a rate of around 20% annually [17]. The Quantum Support Vector Machine (QSVM) [18] is the most popular supervised technique in QML; it leverages the higher-dimensional vector space optimization limit to categorize labeled data. To make the analysis simpler, QSVM eliminates and builds patterns from unlabeled data [19].

The core principle of quantum parallelism is the concurrent storing of multiple qubit states. In addition to the aforementioned, quantum processing is accelerated by entanglement and interference of quantum states. From Feynman's first assertion that quantum computing could effectively simulate complicated quantum systems [20,21], the field has seen fast evolution. In 2014, the first QML application was shown, which included sorting massive volumes of data using QSVM, the SVM algorithm's quantum variant [18]. Along with other learning systems that make use of the quantum operator and superposition, SAL was also developed [22]. Decision tree classification [24] and Quantum K-Nearest Neighbors (QKNN) [23] are two other search approaches that have been proposed.

Support Vector Machines (SVM) and Quantum Support Vector Machines (SVM) are the two algorithms that are used the most often for data categorization in traditional machine learning [17]. In order to address classification issues, a significant variety of classical and quantum support vector machine (SVM) approaches have been researched and established. These methods have been shown in a number of benchmark studies. A comparison of the accuracy and performance of traditional and quantum support vector machine (SVM) methods is shown in [25]. The assessment is made through the application of a quantum support vector machine (QSVM) method with the dataset provided by MNIST of handwritten numbers on the other hand. In order to evaluate the performance of QSVM in comparison to that of SVM, a classification method that is based on QSVM was used to a dataset that included individuals with breast cancer [26]. During the time that they are conducting experiments with noisy quantum computing devices and quantum algorithms in [28], Tang [27] is developing a recommendation system that is based on QML and has the potential to achieve exponential improvement.

To make both conventional and quantum machine learning more precise and efficient, several researchers have proposed novel algorithms and approaches. The breast cancer dataset was used by [29] to create a QSVM algorithm and propose a quantum kernel estimation method that reduces measurement error. Use three simulated datasets to train classification models on quantum and traditional computing backends using QSVM and classical SVM [30]. In [31], the authors evaluate conventional SVMs and QSVMs using suitable quantum feature mapping selection for accuracy and execution performance, using the IBMQ quantum computer. Using support vector machines (SVMs), we may compare conventional and quantum computing, [32] and conduct experiments with huge datasets, like the drug dataset, that are difficult for classical calculations. Lastly, a study by [33] suggests a way to use the SVM algorithm to examine the feature map of a two-qubit kernel-based quantum classifier, and a study by [34] suggests a QLSSVM to address binary classification issues with the cardiac disease dataset.

Numerous general approaches to embedding quantum circuits in ML models are currently available in the literature. To categorize patterns with binary characteristics, a quantum classifier is used to do picture classification [35] utilizing Quantum Machine Learning (QML). Both balanced classification difficulties and unbalanced classes, where the minority class is the most important, have favourable outcomes when tested. In contrast, [36] uses a quantum support vector machine to categorize Big Data. Using continuous variable-based unitary gate operations for binary classification, a Variational Quantum Classifier (VQC) is suggested in [37]. This method, which contains a more effective data representation and is explained in more depth in [38], is important for applications on genuine near-term quantum devices.

In the context of quantum state classification, [39] employs a Tensor Train Network (TTN) variational network to train and classify pure states with different levels of entanglement. According to their findings, while using a hypothetical dataset, it was observed that when the unitary operations of the TTN are too simple,

their categorization accuracy is comparable to that of random class assignments. The TTN can effectively categorize quantum states through the use of advanced methodologies, such as ancilla qubits. A third possible output is associated with an ambiguous conclusion in [40] to replicate the training procedure for Post Quantum Cryptography (PQC) frameworks for classifying quantum states as pure or mixed. In order to replicate the nonlinear behavior of neural networks, their circuits depend on numerous layers of gates which are influenced by measurements.

The suggestion to classify and arrange quantum circuits is an ML challenge that is anticipated to materialize in the real world [41]. They showcase a dataset of quantum circuits that have been tuned using variational quantum eigensolvers. A dataset consisting of 300 samples for each of the six classes of quantum circuits was generated. This dataset was built using ten distinct ansatz, each with depths that ranged from three to thirty-two. Additionally, six common kinds of condensed- matter Hamiltonians were used, each containing four to sixteenth qubits.

The identification of breast cancer is one area where ML and DL approaches are being used to provide more effective tactics for early diagnosis and prediction of this common and potentially deadly illness. [42]. To make their breast cancer predictions, used the Wisconsin Breast Cancer Dataset. Data exploration, Label Encoder, and Normalizer methods were used as pre-processing methodologies before model creation. After dividing the dataset into training and testing sets, several machine-learning approaches were evaluated, with a 96% success rate. These techniques included SVM and Random Forest models. Another tool that the authors used was an ANN model, which reached 99% accuracy and a CNN model that reached 97% accuracy.

A study conducted by [43] demonstrated the effectiveness of QSVM techniques in the diagnosis of BC. In addition to classical models developed using the sci-kit-learn toolbox, their study utilizes IBM quantum processors and quantum simulators. In their comparison to a conventional QSVC approach, the authors tested IBM's quantum processor Quito and the quantum simulator on the same dataset. They found that the quantum simulator can reach a 98% accuracy rate with a 0.1 error rate by using a noise reduction approach. The quantum computer Quito by IBM has an initial accuracy of 95%. Alternatively, the standard QSVC method achieves an accuracy of 85%. This study paves the way for further quantum applications by providing strong evidence that quantum support vector machine techniques are successful. Quantum solutions may be made more robust in noisy environments, according to the scientists, if noise reduction procedures are used. Quantum methods running on NISQ devices outperform traditional ML methods after certain adjustments, according to their results.

Another study showed that quantum neural networks might identify cancer [44]. To determine whether cancer cells were benign or malignant, they used a quantum neural network (QNN) using the Breast Cancer Wisconsin (Diagnostic) dataset. In comparison to CNN, their findings demonstrate that QNNs can achieve higher accuracy with much less computing power. Last but not least, research [45] proposed a hybrid machine-learning strategy for picture categorization that combines quantum and classical principles. Combining a conventional support vector machine with several quantum feature maps and kernels, they conducted experiments using the MNIST dataset. They found that their hybrid technique was more accurate and used fewer resources than traditional machine learning approaches. The aforementioned research concludes that ML and QML techniques are useful in detecting and predicting breast cancer, and they also show how quantum computing may improve upon conventional machine learning approaches.

The review of existing literature [51-53] and related works underscores the significant potential of Quantum Machine Learning (QML) models in enhancing breast cancer detection methodologies. However, a technical analysis reveals discernible research gaps in the deployment of QML algorithms for this purpose. In particular, while the results of simulating and implementing QML classification algorithms with the help of the Qiskit library have been encouraging, there remains a need for comprehensive studies that address the scalability of these models, the optimization of quantum feature selection processes, such as the application of Principal Component Analysis (PCA) in aligning with the limited number of qubits, and the comparative efficiency of quantum versus classical algorithms in real-world computational environments. Furthermore, the literature indicates a lacuna in the evaluation of computational costs associated with quantum models, particularly in terms of quantum resource allocation and the practical feasibility of executing these models on current quantum hardware. Addressing these research gaps could significantly advance the applicability and effectiveness of Applying QML models to breast cancer screening and diagnosis at an early stage, thereby contributing to improved diagnostic outcomes and patient care.

MATERIAL AND METHODS:

3 PROPOSED METHODOLOGY

The proposed methodology for structured quantum machine learning (QML) workflow for breast cancer diagnosis is presented in Figure 1, commencing with dataset preprocessing to optimize data quality. Subsequently, a feature subset is constructed to distil the most informative attributes, followed by partitioning the dataset into training and testing sets.

The crux of the workflow involves encoding these features into qubits using quantum circuits with various feature maps like `ZZFeatureMap`, `RealAmplitudes`, and `EfficientSU2`. These encoded features are then processed using four distinct QML algorithms—Quantum Variational Quantum Classifier (QVQC), Quantum Support Vector Machine (QSVM), Quantum Random Forest (QRF), and Quantum k-Nearest Neighbors (QkNN)—each harnessing quantum states and computations to predict breast cancer occurrences. The final phase encompasses a meticulous performance evaluation and comparative analysis of these models, measuring their predictive accuracy and determining the most effective algorithm for this critical healthcare application.

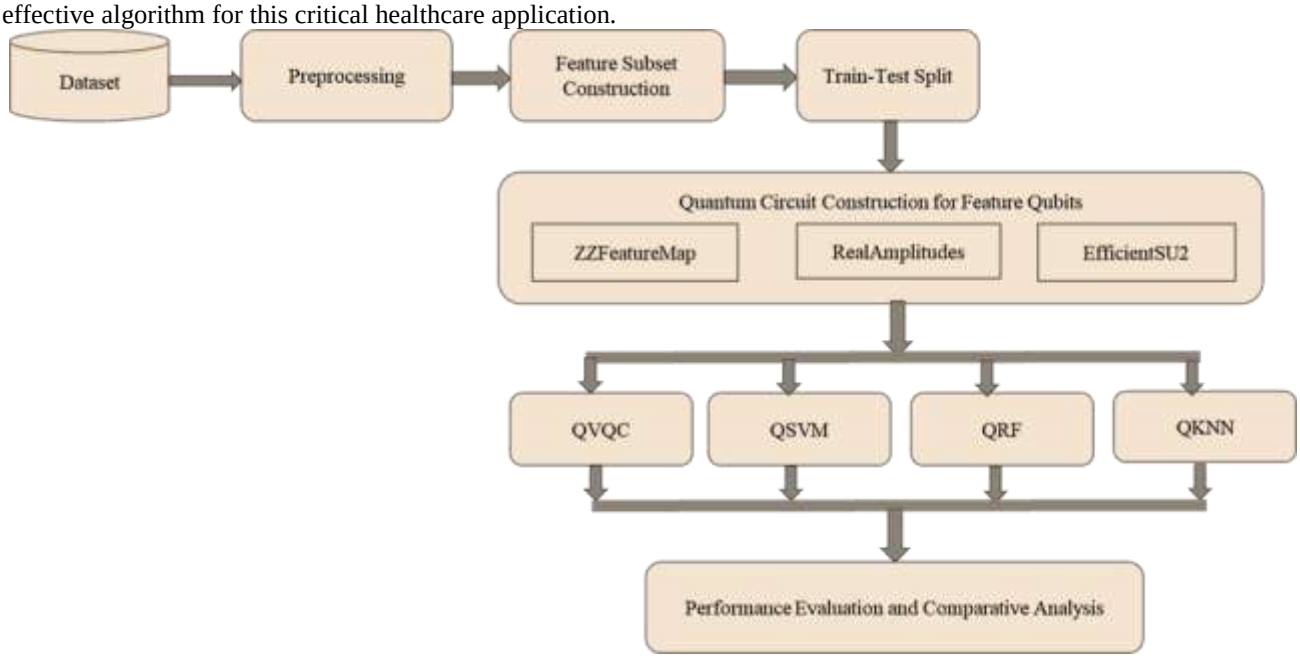

Figure 1: Proposed Methodology

3.1. PREPROCESSING

In the preprocessing stage of the depicted QML workflow for breast cancer diagnosis, the dataset undergoes a series of sophisticated processing techniques to ensure data integrity and readiness for quantum computation. Normalization is applied to rescale the features into a uniform range, preserving relative distributions while aligning them with the operational requirements of quantum algorithms. Concurrently, a thorough missing values analysis is conducted to identify and rectify gaps in the dataset, employing strategies such as imputation or omission to handle incomplete data points effectively. This ensures the dataset's completeness, which is critical for the subsequent quantum encoding processes. Additionally, exploratory data analysis (EDA) is meticulously performed to unearth underlying patterns, detect anomalies, and gain insights into the data's structure and relationships. This analytical exploration involves statistical summaries, correlation assessments, and visualization techniques, which collectively facilitate a deeper understanding of the breast cancer dataset's characteristics, thereby informing the selection of feature subsets and optimizing the application of quantum machine learning algorithms for diagnosis.

3.2. FEATURE ATTRIBUTE CONSTRUCTION

Principal Component Analysis (PCA) is an essential part of the Quantum Machine Learning (QML) methodology for breast cancer diagnostics in terms of extracting features and reducing dimensionality. Principal component analysis (PCA) [19] separates the initial, potentially associated characteristics into a new group of independent variables called PCs. The initial principal component takes into consideration the maximum variance in the data set, and subsequent components have the maximum variance possible while still being orthogonal to the ones before them. The initial variables are linearly combined to form these major components. A mathematical representation of the transformation is as:

$PCi = ai1X1 + ai2X2 + \cdots + ainXn$ (Eq.1)

where PCi is the i-th principal component, Xj is the j-th original feature, and aij is the weight given to Xj for PCi.

The PCs serve as the feature attributes which are then mapped onto qubits in the quantum circuit. In this context, the selected principal components essentially define the state space of the qubits, with each component corresponding to a quantum state that can be prepared and manipulated within the quantum circuit. By encoding these PCs into the quantum domain, we leverage the high-dimensional Hilbert space afforded by quantum systems to perform complex computations that are expected to yield insights for the classification task at hand.

After identifying the principal components with PCA and mapping them onto qubits, the quantum circuit is constructed and subsequently utilized to train the QML models. This involves preparing the initial quantum states corresponding to the PCs, applying quantum gates that evolve these states, and finally measuring the outcome, which relates to breast

cancer diagnosis prediction. The training process includes an optimization routine that adjusts the parameters of the quantum gates, aiming to minimize a cost function reflective of the prediction error, thus enhancing the model's predictive capabilities. The use of PCA-derived features in this quantum framework is a strategic choice that aims to capture the most significant aspects of the dataset while facilitating efficient and potent implementation of QML algorithms for disease diagnosis.

3.3. QUANTUM CIRCUIT DESIGN

A method used in quantum machine learning, quantum circuit creation employing feature maps involves transforming conventional data into a quantum format that quantum computers can handle. This method allows quantum algorithms to efficiently process complicated data structures by mapping aspects of classical data onto quantum states. An essential part of this procedure is feature maps, which transform traditional data into quantum states amenable to processing and manipulation using quantum gates. Improvements in quantum machine learning methods are possible because feature maps enable quantum circuits to process high-dimensional data and execute operations such as pattern recognition, classification, and regression.

3.3.1. ZZFeatureMap

A quantum neural network is built using a feature map and ansatz to train a Qiskit machine learning algorithm Variational Quantum Classifier. It is crucial that the feature mappings accurately encode what is being stored into the qubits utilized, since each normalized feature is allocated one qubit. The feature map has been separated using the ZZFeatureMap preset structure from the Qiskit library [46], as seen in Figure 2, for a case with four variables. This structured approach helps in creating a meaningful and effective quantum model for processing the data and achieving desired classification outcomes.

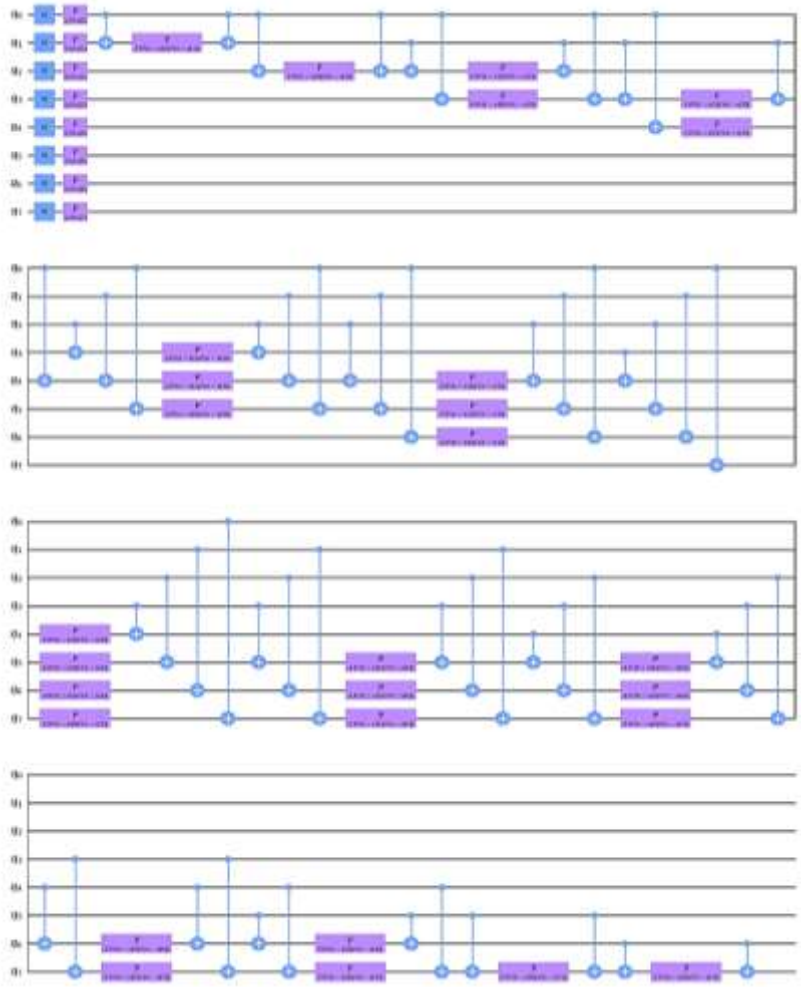

Figure 2: Visualize using the default Qiskit structure Qubits=8 and ZZFeatureMap.

.

3.3.2. Ansatz – RealAmplitude

After constructing the feature map using the ZZFeatureMap feature qubits, the next step is to add a parameterized quantum circuit known as an ansatz with RealAmplitudes [47]. The ansatz circuit contains a set of tunable parameters or

weights that are optimized through the minimization of an objective function. The disparity between the quantum model's predictions and the labeled data is quantified by this objective function. Figure 3, is a diagram of an analogue circuit that can process eight characteristics.

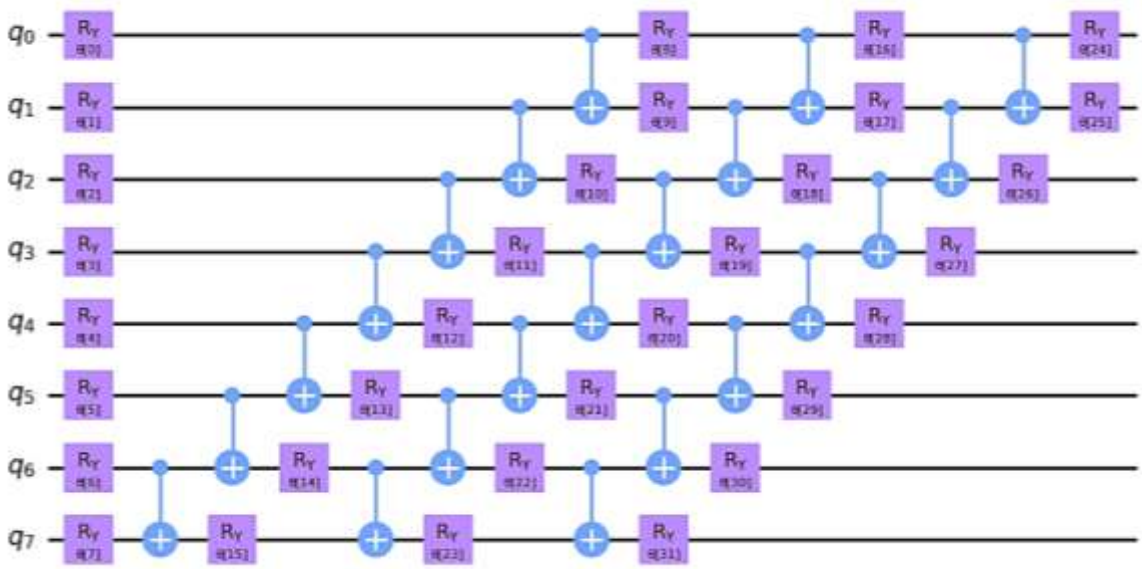

Figure 3: RealAmplitudes ansatz circuit with feature qubits = 8.

3.3.3. Ansatz – EfficientSU2

When building models of quantum algorithms, it is common practice to use a set of predefined gates or quantum operations to train for the best parameterization of those gates. Two preset models, RealAmplitudes and EfficientSU2 [47, 48], were used to create this collection of gates that implement the ansatz. You may see a comparison of the two models' performance in the results section. Heuristic trial wave functions like the RealAmplitudes circuit have many uses, such as chemical ansatzes and machine learning classification circuits. The circuit's layers alternate between CX entanglements and Y rotations. A variety of pre-made entanglement patterns are available, or users may make their own. Since the amplitudes of the prepared quantum states are always real and have a complex component of zero, they are referred to as RealAmplitudes.

The circuit known as EfficientSU2 consists of many layers or single-qubit computations that have connections by SU(2) and CX connections. The members belonging to the specialized unitary group with order 2, denoted as SU(2), consist of the 2x2 unitary matrix of determinant 1. Machine learning classification circuits and variational quantum algorithms' trial wave functions may both be built using this heuristic method. Figure 4, shows the discrepancy compared to the RealAmplitudes circuit model. In addition, the training procedure requires an optimization method. According to the study, the energy evolution of each iteration changes depending on the optimizer used. Since COBYLA shortens the time required to achieve significant progress with each iteration, it is the most effective optimizer. To make the training process faster, we used a gradient-free. When time is of the essence and one does not know the objective function's derivative, the COBYLA numerical optimization method may be used.

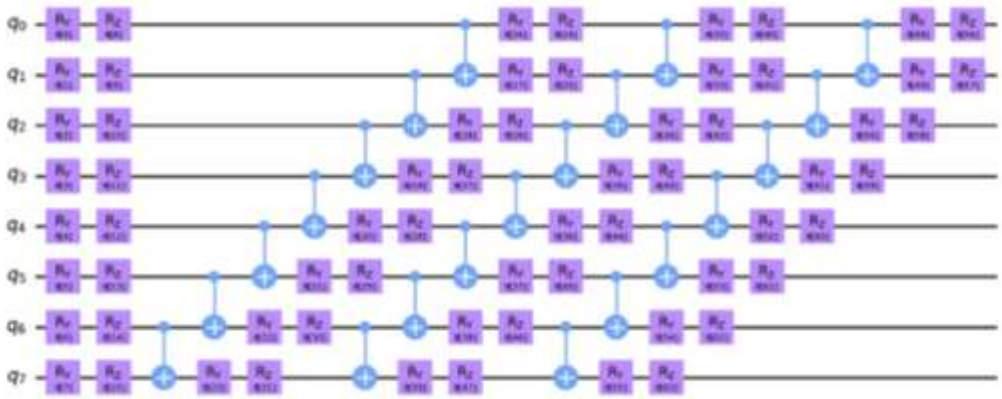

Figure 4. EfficientSU2 ansatz circuit with feature qubits = 8

3.4. CONSTRUCTION OF QUANTUM CLASSIFIER

The construction of a quantum classifier begins with preprocessing the input data, followed by mapping classical features to quantum states using quantum feature maps like ZZFeatureMap. An ansatz circuit is then designed as a parameterized quantum circuit with tunable weights, representing the quantum model's learning ability. These components are combined into a quantum circuit, and the classifier is trained by optimizing the ansatz circuit's parameters to minimize an objective function. Validation and testing ensure the classifier's accuracy, and once trained, the quantum circuit is executed on quantum hardware or simulators to classify new data samples. Various QML classifiers are used to design and assess the predictions and analyse feature importance, completing the construction of an efficient quantum classifier for classification tasks.

3.4.1. VQC Classifier

One kind of quantum circuit that may be trained to categorize data is the Variational Quantum Circuit (VQC). This is achieved by adjusting the gate settings. Quantum classifiers are built using it. A feature map for quantum data encoding, an optimizer for parameter optimization to minimize the cost function, and a variational form (or ansatz) for parameterized circuits are all part of the VQC. This approach, which is comparable to traditional machine learning, may, by using the principles of quantum computing, enhance performance on certain tasks. One such setup includes an optimization procedure, a bespoke ansatz for the variational portion, and a ZZFeatureMap for data encoding.

The ZZFeatureMap is a quantum feature map that makes use of the high-dimensional Hilbert space to effectively encode classical data into a quantum state. Layers of parameterized ZZ gates, which are two-qubit operations on pairs of qubits, enable it to capture correlations between various characteristics of the input data and are used to construct it. The model of a single ZZ gate operating on two qubits may be expressed mathematically as the unitary operator

$$U(\phi) = e^{-i\phi Z \otimes Z} \qquad \text{(Eq.2)}$$

where ϕ is a feature-related parameter and Z is the Pauli-Z matrix.

First, **the ansatz**, also known as a variational circuit, is a parameterized quantum circuit that may search for a solution by probing the space of possible quantum states. The optimization method involves adjusting the settings of a series of quantum gates, which is its typical structure. The VQC's efficacy and success are very sensitive to the ansatz's shape. A typical layout involves entangling gates and single-qubit rotating gates stacked on top of each other. One way to depict the rotating gates is as

$$Rx(\theta) = e^{-i\theta X/2} \qquad \text{(Eq.3)}$$

$$Ry(\theta) = e^{-i\theta Y/2} \qquad \text{(Eq.4)}$$

$$Rz(\theta) = e^{-i\theta Z/2} \qquad \text{(Eq.5)}$$

Where X,Y, and Z are Pauli matrices.

Optimizer: The cost function, usually the discrepancy between the training data's actual labels and the anticipated labels, is what the optimizer is trying to reduce. By adjusting the ansatz's parameters, the optimization process finds the cost function's minimum. Gradient descent, the COBYLA optimizer, and the Quantum Natural Gradient are examples of renowned quantum-classical hybrid optimization methods.

Cost Function: The cost function quantifies the performance of the VQC on the initial training data. A commonly used approach involves using the projected amount of the measurement operators that is linked to the outcomes of the classification process. The cost of a function is expressed mathematically as

$$C(\ddot{\Theta}) = \langle \boldsymbol{\psi}(\ddot{\Theta}) \mid \boldsymbol{M} \mid \boldsymbol{\psi}(\ddot{\Theta}) \rangle \qquad \text{(Eq.6)}$$

The VQC-prepared quantum state with parameters $\ddot{\Theta}$, and M is the measurement operator associated with the classification job.

Training Loop: During training, the optimizer repeatedly adjusts the ansatz's parameters according to the cost function's gradient until a stopping threshold is reached, such as a minimum change in the cost function or a maximum number of iterations

Callback Function: A callback function, like **callback_graph**, can be used to monitor the training process, allowing for real-time visualization or logging of metrics such as the value of the cost function or the parameters at each iteration

In constructing a VQC for classification with a ZZFeatureMap and a custom ansatz, the overall goal is to iteratively update the parameters of the ansatz to map the input features to the correct output labels, using entanglement and superposition, two quantum mechanical features, to improve performance than classical counterparts on certain tasks.

Algorithm 1: Training Variational Quantum Classifier (VQC)

1. Import necessary libraries: import time, import numpy as np.
2. Import the VariationalQuantumClassifier module from your_custom_module (assuming it exists and is correctly implemented).

3. Set configuration parameters for VQC: optimizer (quantum optimizer for training), feature_map (quantum feature map), ansatz (quantum ansatz circuit), callback_graph (callback function for optimization), and other hyperparameters as needed.
4. Prepare the training data: Convert your training features and labels from pandas DataFrame to numpy arrays (train_features and train_labels).
5. Initialize the Variational Quantum Classifier (vqc) with the specified configuration parameters.
6. Optionally, clear the objective function value history if applicable by defining objective_func_vals as an empty list.
7. Measure training time:
   a. Start the timer: start_time = time.time()
   b. Fit the Variational Quantum Classifier on the training data: vqc.fit(train_features, train_labels)
   c. Calculate the elapsed time: elapsed_time = time.time() - start_time
8. Output the training time: Print the training time in seconds using the elapsed_time variable.

3.4.2. QSVM classifier

A Quantum Support Vector Machine (QSVM) classifier is built by combining conventional SVM with quantum computing to improve its hyperplane detecting capabilities. The quantum feature map is responsible for converting classical data into quantum states in a high-dimensional Hilbert space. This allows for the use of quantum superposition and entanglement, and it is the core component of a QSVM. This transformation often results in a feature space where complex, nonlinear patterns become linearly separable. Training and classification are the two main components of the QSVM algorithm. The training process begins with the quantum feature map encoding the input into quantum states. Subsequently, the kernel matrix is calculated to capture the inner products of these states. A traditional support vector machine (SVM) approach takes this quantum-enhanced kernel matrix and uses it to solve a convex optimization problem to find the best separation hyperplane. The support vector machine (SVM) can be used in a feature space with a high degree of dimension by concealing the calculation of data coordinates and using the kernel approach to compute the similarities between data points. Quantum Support Vector Machines (QSVM) use the quantum microprocessor to enhance the efficiency of kernel matrix calculations for certain data types, hence possibly providing exponential speedup compared to conventional computing for particular datasets. The kernel equation in a QSVM may be mathematically represented as

$K(x_i,x_j) = <\phi(X_i)|\phi(X_j)>$ (Eq.7)

where $\phi(X_i)$ is the quantum feature map applied to data point X_i and < | denotes the inner product in the Hilbert space. This approach marries the robustness of SVMs with the computational advantages of quantum information processing, aiming to achieve superior classification accuracy and effectiveness on feature spaces increased by quantum mechanics.

Algorithm 2: Training Quantum Support Vector Machine (QSVM)

1. Import necessary libraries: import time, import numpy as np.
2. Import the QuantumSVMClassifier module from your_custom_module (assuming it exists and is correctly implemented).
3. Set configuration parameters for QSVM: kernel_type (type of kernel for SVM, e.g., 'linear', 'rbf'), C (regularization parameter for SVM), quantum_instance (quantum backend or simulator), optimizer (quantum optimizer for training), and callback_graph (callback function for optimization).
4. Prepare the training data: Convert your training features and labels from pandas DataFrame to numpy arrays (train_features and train_labels).
5. Initialize the Quantum Support Vector Machine Classifier (qsvm) with the specified configuration parameters.
6. Optionally, clear the objective function value history if applicable by defining objective_func_vals as an empty list.
7. Measure training time:
   a. Start the timer: start_time = time.time()
   b. Fit the Quantum Support Vector Machine Classifier on the training data: qsvm.fit(train_features, train_labels)
   c. Calculate the elapsed time: elapsed_time = time.time() - start_time
8. Output the training time: Print the training time in seconds using the elapsed_time variable.

3.4.3. QRF Classifier

The Quantum Random Forest (QRF) classifier described is tailored for a quantum-enhanced approach to classification tasks, harnessing the power of quantum computing to potentially outperform classical machine learning algorithms. This QRF setup involves the creation of an ensemble of three quantum decision trees, each equipped with a dynamic choice among three types of quantum ansatz: `eff_anz_pqc_arch`, `iqp_anz_pqc_arch`, and `eff_anz_pqc_arch`. These ansatz options allow for a versatile approach to quantum state preparation and manipulation down different paths of the decision trees, catering to the specific demands of the dataset being analyzed.

The quantum feature embedding for this classifier utilizes a total of 6 qubits, with the type of embedding (`as_params_all`, `as_params_iqp`, `as_params_all`) selected to match the chosen ansatz for each level of the tree, ensuring optimal data representation in the quantum domain. The QRF operates with a specific focus on maximizing

information gain (`SplitCriterion.init_info_gain('clas')`) as the criterion for splitting nodes, a strategy aimed at enhancing the classifier's accuracy by ensuring that each split contributes significantly to class distinction.

To estimate the kernel in the quantum decision trees, the setup involves taking 2024 circuit samples per kernel estimation. The model supports classification tasks with a variable number of classes (`num_classes`) and is designed to explore the data space efficiently up to a maximum tree depth of 4. The complexity of the quantum embedding is reflected in the number of parameters (`num_params_split`), which varies according to the chosen ansatz and is calculated as $n$qubits×($n$qubits+1)for this configuration.

To facilitate effective training and testing processes, the QRF utilizes landmarks (a reduced subset of the training data) with a set number of 5 (`svm_num_train`) for constructing the SVM within the trees, applying a regularization term (`svm_c`) to balance model complexity and training error minimization. The model is fine-tuned to ensure that splits occur only when they involve at least as many samples as there are landmarks, thereby maintaining robustness and generalization capability.

The QRF classifier is built to run on the `cirq` quantum simulation platform, with computational efficiency bolstered by parallel processing capabilities set to utilize three cores (`cores = 3`). This parallelism is reflected in the training phase, where the ensemble is trained on a given training set with a partition sample size of 180, and can be toggled off (`parallel=False`) during testing for compatibility with additional diagnostic measures such as tree correlation analysis (`calc_tree_corr=True`).

The described setup showcases the nuanced application of quantum computing principles to the task of data classification, leveraging the high-dimensional feature spaces accessible through quantum state manipulation to potentially achieve enhanced classification performance. This Quantum Random Forest model embodies the cutting-edge integration of quantum and classical computing techniques, aiming to unlock new capabilities in the field of machine learning.

Algorithm 3: Quantum Random Forest Classification

1. Import necessary libraries and modules.
2. Define configuration parameters including the number of qubits, branching variants, number of trees, sample numbers, class numbers, maximum tree depth, parameters for splitting nodes, SVM regularization parameters, embedding types, and CPU cores.
3. Initialize the Quantum Random Forest Classifier with the specified configuration parameters.

```
qrf = QuantumRandomForestClassifier(
    n_qubits=n_qubits,
    branch_var=branch_var,
    num_trees=num_trees,
    pqc_sample_num=pqc_sample_num,
    num_classes=num_classes,
    max_depth=max_depth,
    num_params_split=num_params_split,
    svm_c=svm_c,
    embedding_type=embedding_type,
    cores=cores
```

4. Utilize the load_dataset function to import the set of training and testing data, specifying the dataset name.
5. Utilize the training set and labels to learn the Quantum Random Forest classifier, while selecting the sample size for the partition.
6. Test the trained classifier with the testing set and labels, optionally returning predictions and calculating tree correlations.
7. Output the test accuracy and, if applicable, the tree correlations between different trees in the forest.

3.4.4. QkNN Classifier

One quantum-enhanced variant of the traditional k-Nearest Neighbors technique is the Quantum k-Nearest Neighbors (QkNN) classifier, utilizing quantum computing to potentially expedite and improve the classification process. This implementation begins by importing necessary libraries, including `time` for benchmarking and `numpy` for numerical operations, alongside a custom module that houses the `QuantumKNeighborsClassifier`. The classifier is initialized with key parameters such as the number of neighbors (`n_neighbors`), a quantum `sampler`, a `feature_map` a "prototype" for parameterized quantum circuits, a method for incorporating classical data into quantum states, and an `optimizer` to fine-tune the parameters. Additionally, a `callback_graph` function is specified for real-time monitoring of the optimization process.

To ensure a clean starting point for optimization, any existing history of objective function values is cleared. The training data, potentially originating from pandas data structures, is converted into numpy arrays to comply with the quantum processing requirements. The fitting process is then initiated, marking the start time before invoking the `fit` method of the QkNN classifier with the prepared training features and labels. Upon completion, the elapsed training time is calculated and reported, providing insight into the computational efficiency of the quantum approach.

This QkNN setup leverages quantum computing's ability to process information in parallel and access high-dimensional feature spaces, potentially offering significant advantages over traditional alternatives regarding precision and velocity. By incorporating quantum mechanics into the k-Nearest Neighbors framework, this classifier aims to harness the unique properties of quantum states and operations to enhance the machine learning pipeline, from data encoding and model training to the optimization of hyperparameters.

Algorithm 4: Training Quantum k-Nearest Neighbors Classifier

1. Import necessary libraries: import time, import numpy as np.
2. Import the custom module QuantumKNeighborsClassifier from your_custom_module.
3. Set configuration parameters for QkNN: n_neighbors (number of neighbors), sampler (quantum sampler), feature_map (quantum feature map), ansatz (quantum ansatz), optimizer (quantum optimizer), and callback_graph (callback function for optimization).
4. Prepare the training data: Convert your training features and labels from pandas DataFrame to numpy arrays (train_features and train_labels).
5. Initialize the Quantum k-Nearest Neighbors Classifier (qknn) with the specified configuration parameters.
6. Optionally, clear the objective function value history if applicable by defining objective_func_vals as an empty list.
7. Measure training time:
   a. Start the timer: start_time = time.time()
   b. Fit the Quantum k-Nearest Neighbors Classifier on the training data: qknn.fit(train_features, train_labels)
   c. Calculate the elapsed time: elapsed_time = time.time() - start_time
8. Output the training time: Print the training time in seconds using the elapsed_time variable.

RESULTS AND DISCUSSIONS:

In our study, we have extended our exploration into the realm of Quantum Machine Learning (QML) by employing IBM Qiskit Aqua, a comprehensive library tailored for devising and implementing quantum algorithms. This framework is particularly noted for its suite of pre-built quantum machine learning algorithms, facilitating our investigation into advanced computational techniques. An important area of medical diagnostics, our research involves applying four quantum machine learning algorithms—Quantum Support Vector Machine (QSVM), Variational Quantum Classifier (VQC), Quantum Random Forest (QRF), and Quantum k-Nearest Neighbors (QkNN)—to the diagnosis of breast cancer.

For our experiments, the direct kernel-based method for Support Vector Machine (SVM) classification was executed on a classical computer leveraging a standard CPU. Additionally, we have implemented the QRF and QkNN algorithms, extending our quantum computational approach to these sophisticated classification techniques. It is imperative to note, however, that all quantum machine learning algorithms in this study were simulated, leveraging the computational capabilities of an i5 processor equipped with 16 GB of RAM, and utilizing the robust IBM Qiskit packages. This simulation-based approach inherently introduces limitations to our comparison, as the quantum algorithms do not run on actual quantum hardware, potentially affecting the fidelity and efficiency of the results. Nonetheless, this methodology allows us to critically assess the theoretical performance and practical feasibility of applying quantum computing to the nuanced domain of breast cancer diagnosis, marking a significant step forward in the intersection of quantum computing and healthcare.

4.1. DATASET DESCRIPTION

The study focuses on breast cancer, which originates in mammary cells and is characterized by abnormal cell growth, rapid division, and potential metastasis to other parts of the body [49]. The research aims to distinguish between benign and malignant breast tissue samples. The dataset used for analysis contains information on breast features indicative of tumors, in particular those that are benign or cancerous. Cell nuclei properties such as radius, texture, area, smoothness, compactness, concavity, concave points, symmetry, and fractal dimension are described by these features, which are extracted from digital pictures of Fine Needle Aspirate (FNA) samples of breast masses. Thirty characteristics are produced by calculating the mean, standard error, and worst value of the three biggest measures. The dataset is normalized using min-max scaling to ensure data homogeneity and any null or unobserved values are removed from the dataset.

4.2. PERFORMANCE MEASURES

In the evaluation of models in this research, various relevant metrics were applied to assess their performance [50]. Accuracy, serving as a crucial metric, quantifies the model's overall performance by measuring the fraction of instances correctly classified. Precision indicates the fraction of correct positive predictions, while recall represents the fraction of actual positives predicted correctly. The F1-Score balances between recall and precision.

$$TPR = \frac{TP}{TP+FN} \quad \text{(Eq.8)}$$

$$FPR = \frac{FP}{TP+FN} \quad \text{(Eq.9)}$$

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad \text{(Eq.10)}$$

$$precision = \frac{TP}{TP+FP} \quad \text{(Eq.11)}$$

$$\text{Re}call = \frac{TP}{TP+FN} \quad \text{(Eq.12)}$$

$$F1-Score = \frac{2*(precision*\text{Re}call)}{precision+\text{Re}call} \quad \text{(Eq.13)}$$

4.3. PERFORMANCE ON VQC ON BREAST CANCER DATASET.

The performance evaluation of the Variational Quantum Classifier (VQC) on the Breast Cancer dataset showcases its efficacy in accurately distinguishing between benign and malignant breast tissue samples. The table 1, presents The Variational Quantum Classifier (VQC) applied to the breast cancer dataset exhibits a discernible performance trend relative to the number of feature qubits, with the highest accuracy of 0.90 for both training and testing phases achieved with a 4-qubit feature map. This configuration indicates an effective balance between complexity and predictive capability within the ansatz feature map. As the model complexity increases to 6 and then 8 feature qubits, a gradual decline in accuracy is observed, falling to 0.84 and 0.82 for training and testing respectively with 6 qubits, and further to 0.79 and 0.77 with 8 qubits. These results imply that larger feature spaces may introduce redundancy or overfitting, thereby reducing the model's effectiveness, and highlighting the necessity of optimizing the ansatz complexity to maintain an efficient and accurate quantum classification model for this dataset. The figure 5, represents the data distribution with ZZFeatureMap using VQC classifier.

Table 1: Variational Quantum Classifier (VQC) applied to the breast cancer dataset for different feature qubits

Model	Train Score	Test Score
VQC with 4 Features qubits	0.90	0.90
VQC with 6 Features qubits	0.84	0,82
VQC with 8 Features qubits	0.79	0.77

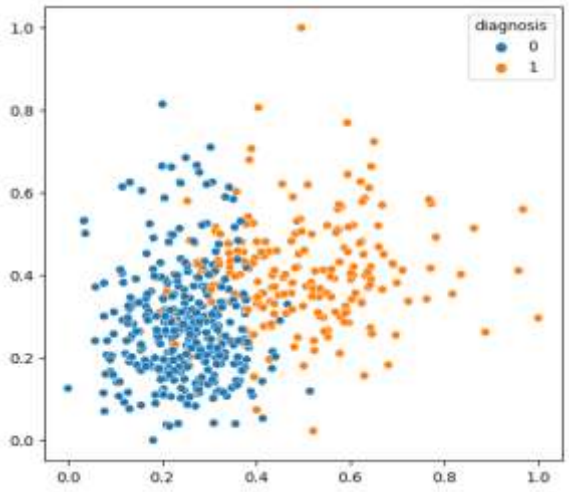

Figure 5: The graph that is produced by utilizing RealAmplitudes in VQC with 8 feature qubits.

4.4. PERFORMANCE ON QSVM ON BREAST CANCER DATASET.

The evaluation of the Quantum Support Vector Machine (QSVM) on the Breast Cancer dataset demonstrates its effectiveness in accurately classifying breast tissue samples as benign or malignant, highlighting its potential for improving diagnostic accuracy in cancer detection.

Table 2: QSVM applied to the breast cancer dataset for different feature qubits

Model	Train Score	Test Score
QSVM with 4 Features qubits	0.89	0.89
QSVM with 6 Features qubits	0.84	0.84
QSVM with 8 Features qubits	0.74	0.71

Table 2, the Quantum Support Vector Machine (QSVM) shows varying performance on the breast cancer dataset, which seems to be inversely related to the number of feature qubits utilized. With 4 feature qubits, the QSVM model demonstrates a robust performance, achieving a train and test score of 0.89, suggesting that the quantum feature map at this level is sufficiently rich to capture the critical patterns of the data without overcomplicating the model. When the feature space is expanded to 6 qubits, there is a slight decrease in performance, with both the train and test scores dropping to 0.84. This could indicate the beginning of a trade-off between the complexity of the feature map and the model's ability to generalize. Expanding further to 8 feature qubits leads to a more substantial drop in performance, with train and test scores of 0.74 and 0.71 respectively, which may reflect an over-parameterized model that struggles to capture the underlying structure of the dataset efficiently. These results underscore the critical importance of selecting an optimal number of feature qubits in QSVM to balance the expressiveness of the quantum feature map and the generalization capability of the model.

4.5. PERFORMANCE ON QRF ON BREAST CANCER DATASET.

The performance analysis of Quantum Random Forest (QRF) on the breast cancer dataset highlights its effectiveness in accurately classifying breast tissue samples into benign and malignant categories. By harnessing quantum computing principles, QRF demonstrates potential improvements in classification accuracy and robustness compared to classical Random Forest algorithms, thereby contributing to enhanced diagnostic capabilities in breast cancer prediction.

Table 3: QRF applied to the breast cancer dataset for different feature qubits

Model	Train Score	Test Score
QRF with 4 Features qubits	0.91	0.90
QRF with 6 Features qubits	0.88	0.87
QRF with 8 Features qubits	0.81	0.81

The Quantum Random Forest (QRF) classifier's performance on the breast cancer dataset demonstrates a decrease in accuracy with an increasing number of feature qubits as shown in Table 3. When utilizing a 4-qubit feature map, the QRF achieves excellent train and test scores of 0.91 and 0.90, respectively, indicating that a modest quantum state space suffices for capturing the salient features necessary for diagnosis. Expanding the feature map to 6 qubits, the model's accuracy moderately declines to train and test scores of 0.88 and 0.87, suggesting a potential introduction of less relevant features that slightly obscure the model's predictive clarity. A further extension to 8 feature qubits results in train and test scores of 0.81, revealing a more significant impact due to possible overfitting or inclusion of unnecessary complexity, which does not contribute to, and might even hinder the model's performance. This pattern indicates that while additional qubits provide a larger feature space, they may not correspond to increased diagnostic accuracy, emphasizing the need for careful feature map optimization in QRF models for breast cancer detection.

4.6. PERFORMANCE ON QKNN ON BREAST CANCER DATASET.

The performance evaluation of Quantum k-Nearest Neighbors (QkNN) on the breast cancer dataset demonstrates its capability to classify breast tissue samples as benign or malignant with high accuracy. By leveraging quantum computing principles, QkNN offers a promising approach for improving the accuracy and efficiency of breast cancer prediction compared to classical k-Nearest Neighbors algorithms.

Table 4: QkNN applied to the breast cancer dataset for different feature qubits

Model	Train Score	Test Score
QkNNF with 4 Features qubits	0.87	0.86
QkNN with 6 Features qubits	0.76	0.76
QkNN with 8 Features qubits	0.69	0.68

For the Quantum k-Nearest Neighbors (QkNN) model applied to breast cancer diagnosis, a clear trend emerges where the performance diminishes as the number of feature qubits increases. From table 4, the highest level of accuracy is observed with 4 feature qubits, achieving a train score of 0.87 and a test score of 0.86. This suggests that a quantum feature map with 4 qubits is optimal in this case, providing a rich enough representation of the data for effective classification without overcomplicating the model. When the model is expanded to include 6 feature qubits, both the train and test scores drop

to 0.76, implying that the increased dimensionality of the feature space may start to incorporate noise or irrelevant features that degrade the model's predictive power. Further expansion to 8 feature qubits results in train and test scores of 0.69 and 0.68, respectively, reinforcing the trend where a larger quantum state space leads to reduced accuracy. This decline could be attributed to the challenges of managing a more complex feature space, which can exacerbate issues like quantum noise and hinder the model's ability to generalize from the training data. These findings suggest that for the QkNN model on this breast cancer dataset, a carefully selected feature space that avoids overfitting and quantum noise is key to maintaining high classification performance.

4.7. QUANTUM MACHINE LEARNING MODEL PERFORMANCE ON BREAST CANCER DATASET.

The performance metrics of various Quantum Machine Learning (QML) models on the breast cancer dataset reveal a spectrum of effectiveness across different evaluation criteria. From the Table 5, The Variational Quantum Classifier (VQC) and Quantum Support Vector Machine (QSVM) both achieve the highest accuracy at 0.91. However, QSVM slightly outperforms VQC in terms of precision (0.90 versus 0.88), recall (0.90 versus 0.89), and F1-score (0.90 versus 0.89), suggesting that QSVM may be more effective in both the relevance of its positive predictions and the proportion of correct identifications. The Quantum Random Forest (QRF) shows a modest decline in all metrics with an accuracy, precision, and recall of 0.88, and a slightly lower F1-score of 0.87, indicating a small reduction in the harmonic balance between precision and recall. The Quantum k-Nearest Neighbors (QkNN) model displays the lowest scores among the evaluated models, with an accuracy of 0.86 and corresponding precision and recall of 0.85, leading to an F1-score of 0.86. These scores suggest that while QkNN is less precise and slightly less comprehensive in capturing all relevant instances than its counterparts, it still holds a reasonably good balance as reflected by the F1-score. Collectively, these results underscore the competitive performance of quantum algorithms in analyzing complex biomedical datasets, with QSVM standing out in this instance for its slightly superior balance of accuracy and precision-recall trade-off.

Table 5: QML model performances on breast cancer dataset

QML Model	Accuracy	Precision	Recall	F1-Score
VQC	0.91	0.88	0.89	0.89
QSVM	0.91	0.90	0.90	0.90
QRF	0.88	0.88	0.88	0.87
QKNN	0.86	0.85	0.85	0.86

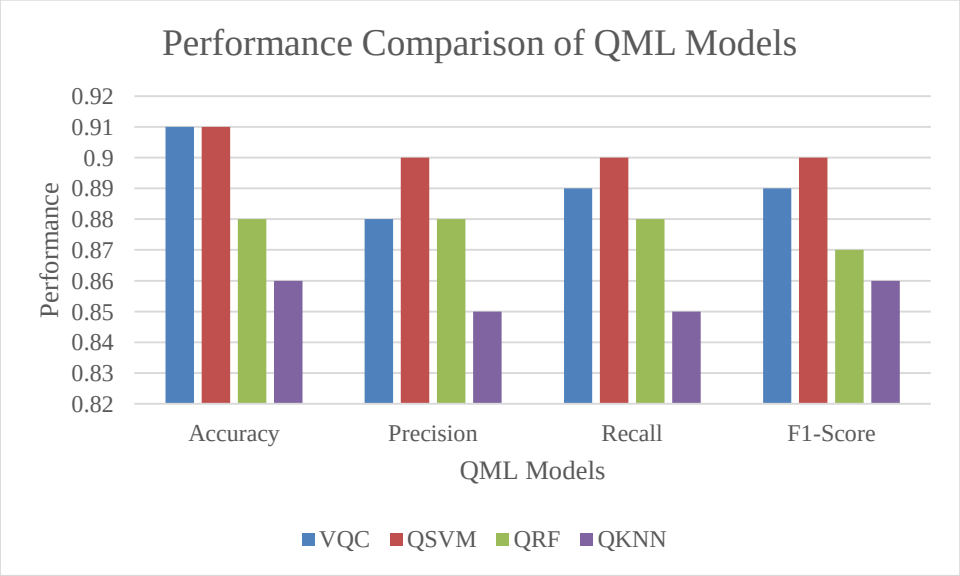

Figure 6: Performance Comparison QML model performances on breast cancer dataset

The figure 6, provides a visual comparison of the performance metrics for four Quantum Machine Learning (QML) models—VQC, QSVM, QRF, and QkNN—tested on a breast cancer dataset. The bar chart illustrates the accuracy, precision, recall, and F1-score for each model, with the y-axis representing performance on a scale from approximately 0.82 to 0.92. Each QML model is denoted by a distinct color. The VQC and QSVM models demonstrate the highest accuracy and precision, closely matched by their recall and F1-scores, indicating well-rounded performance. The QRF model shows slightly lower metrics across all categories but remains competitive. The QkNN model, while having the lowest values among the four, still maintains performance metrics above 0.85 across all evaluated criteria. Overall, the chart suggests that while all models perform well, there are nuanced differences in their predictive capabilities, with VQC and QSVM slightly outperforming QRF and QkNN on this dataset.

4.8. COMPARISON OF QML WITH CLASSICAL MACHINE LEARNING METHODS ON BREAST CANCER PREDICTION

QML methods are compared with classical Machine Learning (ML) techniques for breast cancer prediction, evaluating their respective performances in terms of accuracy, efficiency, and scalability. This comparison aims to assess the potential advantages of leveraging quantum computing principles in medical diagnosis, particularly in distinguishing between benign and malignant breast tissue samples with higher accuracy and computational speed compared to traditional ML approaches.

Table 6: Comparison of QML with classical Machine Learning methods on breast cancer prediction

Quantum vs Classical	ML models	Train Accuracy	Test Accuracy
Quantum ML Models	VQC	0.90	0.90
	QSVM	0.89	0.89
	QRF	0.91	0.90
	QNN	0.87	0.86
Classical ML models	SVC	0.96	0.96
	RF	1.00	0.95
	KNN	0.95	0.92

The comparison between Quantum Machine Learning (QML) models and classical Machine Learning (ML) techniques for breast cancer prediction reveals distinct performance characteristics across both paradigms are shown in Table 6. Within the quantum domain, the Quantum Random Forest (QRF) demonstrates the highest training accuracy at 0.91 and a test accuracy close behind at 0.90, showcasing its robustness and effectiveness in handling the dataset. The Variational Quantum Classifier (VQC) and Quantum Support Vector Machine (QSVM) both exhibit strong performance as well, with identical train and test accuracies of 0.90 and 0.89, respectively, indicating their capability in accurately modeling the breast cancer prediction problem. The Quantum Neural Network (QNN) trails slightly with a train accuracy of 0.87 and a test accuracy of 0.86, still presenting a commendable performance considering the complexities of quantum computation.

On the classical side, the models demonstrate higher accuracies overall, with the Support Vector Classifier (SVC) and Random Forest (RF) leading the pack. SVC achieves both train and test accuracies of 0.96, showcasing its efficiency in linear separation with high-dimensional data. RF achieves a perfect training accuracy of 1.00 and a very high test accuracy of 0.95, reflecting its capacity to capture complex relationships in the data without overfitting, as indicated by the slight drop in test accuracy. The K-Nearest Neighbors (KNN) model also performs admirably, with train and test accuracies of 0.95 and 0.92, respectively, underscoring its effectiveness in classification through a simple yet powerful approach.

This comparison highlights the promising potential of QML models in achieving competitive accuracies with classical counterparts, despite the inherent challenges and the current developmental stage of quantum computing. While classical models currently show higher accuracy, the quantum models are not far behind, suggesting that with further advancements in quantum technology and algorithm optimization, QML models could rival or even surpass classical methods in certain scenarios, offering unique advantages such as computational speedups and handling complex data patterns in high-dimensional spaces.

DISCUSSION

The exploration of Quantum Machine Learning (QML) models for breast cancer diagnosis demonstrates the burgeoning potential of quantum algorithms in tackling complex biomedical datasets. The employment of IBM Qiskit Aqua for simulating QSVM, VQC, QRF, and QkNN algorithms signifies an important step toward integrating quantum computing with healthcare analytics. While the simulated environment, underscores the infancy of practical quantum computing applications, it offers a glimpse into the future where quantum algorithms may revolutionize data-driven diagnostics.

The dataset, centered around the diagnosis of breast cancer through analysis of cell nuclei characteristics from FNA images, serves as a suitable challenge for assessing the efficacy of both quantum and classical ML models. The quantum models, especially QRF and QSVM, show commendable performance, closely rivaling that of their classical counterparts. This suggests that even within the constraints of simulation, QML algorithms possess the capability to discern intricate patterns in data indicative of malignant or benign neoplasia.

However, the comparison with classical ML models reveals a notable gap in accuracy, precision, recall, and F1-Score, with classical algorithms such as SVC and RF showcasing superior performance. This discrepancy highlights the current limitations of QML models, primarily attributed to the challenges of feature space representation and the complexity of quantum state management over larger datasets.

CONCLUSION

Finally, we have shed light on the promise of quantum algorithms in medical diagnostics through our comparative empirical assessment of QML models for breast cancer diagnosis. The use of Quantum k-Nearest Neighbors (QkNN), Quantum Random Forest (QRF), Variational Quantum Classifier (VQC), and Quantum Support Vector Machine (QSVM) within the IBM Qiskit framework demonstrated encouraging outcomes in precisely categorizing breast cancer tissues using feature attributes derived from Fine Needle Aspirate (FNA) images. Notably, QRF and QSVM demonstrated competitive performance levels, with QRF achieving the highest training and testing accuracies among the quantum algorithms studied. This research contributes to the growing body of evidence supporting the integration of quantum computing into medical diagnostics. While QML models show promise, further development and refinement are necessary to achieve parity with classical methods. The research work underscores the importance of ongoing advancements in quantum computing hardware and algorithm optimization to bridge the close the gap in performance and make full use of quantum algorithms' healthcare potential.

Future work can be extended by integrating deep learning concepts with quantum computing to enhance breast cancer prediction. This integration holds promise in leveraging the strengths of both domains, leading to more accurate and robust predictive models for diagnosing breast cancer. Collaborative research efforts between experts in deep learning, quantum computing, and healthcare analytics can propel innovation and provide innovative solutions to advance personalized treatment in cancer, and improve patient outcomes.

REFERENCES

[1]. M. Schuld, I. Sinayskiy, and F. Petruccione. “The quest for a quantum neural network.” Quantum Information Processing, 13(11):2567–2586, 2014.
[2]. J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd. “Quantum machine learning.” Nature, 549(7671):195–202, 2017.
[3]. Siegel RL, Miller KD, Wagle NS, Jemal A. Cancer statistics, 2023. CA Cancer J Clin. 2023;73(1):17-48.
[4]. Waks AG, Winer EP. Breast cancer treatment: a review. Jama. 2019;321(3):288–300.
[5]. Gupta S, Kumar D, Sharma A. Data mining classification techniques applied for breast cancer diagnosis and prognosis. Indian J Comput Sci Eng (IJCSE). 2011;2(2):188–95.
[6]. Salama GI, Abdelhalim M, Zeid MA-E. Breast cancer diagnosis on three different datasets using multi-classifiers. Breast Cancer (WDBC). 2012;32(569):2.
[7]. Paparo GD, Dunjko V, Makmal A, Martin-Delgado MA, Briegel HJ. Quantum speedup for active learning agents. Phys Rev X. 2014;4(3):031002.
[8]. Murty MN, Raghava R. “Kernel-based SVM.” Support vector machines and perceptrons. Cham: Springer; 2016. p. 57–67
[9]. Mangasarian OL, Musicant DR. Lagrangian support vector machines. J Mach Learn Res. 2001;1(Mar):161–77.
[10]. Biamonte J, Wittek P, Pancotti N, Rebentrost P, Wiebe N, Lloyd S. Quantum machine learning. Nature. 2017;549(7671):195–202.
[11]. Rebentrost P, Mohseni M, Lloyd S. Quantum support vector machine for big data classification. Phys Rev Lett. 2014;113(13):130503
[12]. Ruan, Y.; Xue, X.; Liu, H.; Tan, J.; Li, X. Quantum algorithm for k-nearest neighbors classification based on the metric of hamming distance. Int. J. Theor. Phys. 2017, 56, 3496–3507.
[13]. Zhang, Y.; Ni, Q. Recent advances in quantum machine learning. Quantum Eng. 2020, 2, e34.
[14]. Qiskit 0.42.0 Documentation. Available online: https://qiskit.org/documentation/index.html
[15]. Karamizadeh, S.; Abdullah, S.M.; Manaf, A.A.; Zamani, M.; Hooman, A. An overview of principal component analysis. J. Signal Inf. Process. 2020, 4, 173.
[16]. Hu, W. Comparison of Two Quantum Nearest Neighbor Classifiers on IBM’s Quantum Simulator. Nat. Sci. 2018, 10, 87–98.
[17]. Biamonte, J.; Wittek, P.; Pancotti, N.; Rebentrost, P.; Wiebe, N.; Lloyd, S. Quantum Machine Learning. Nature 2017, 549, 195–202.
[18]. Rebentrost, P.; Mohseni, M.; Lloyd, S. Quantum Support Vector Machine for Big Data Classification. Phys. Rev. Lett. 2014, 113, 130503.
[19]. Abdi, H.; Williams, L.J. Principal Component Analysis: Principal Component Analysis. WIREs Comp. Stat. 2010, 2, 433–459.
[20]. Duan, B.; Yuan, J.; Yu, C.-H.; Huang, J.; Hsieh, C.-Y. A Survey on HHL Algorithm: From Theory to Application in Quantum Machine Learning. Phys. Lett. A 2020, 384, 126595.
[21]. Feynman, R.P. Feynman Lectures on Computation, 1st ed.; Hey, T., Allen, R.W., Eds.; CRC Press: Boca Raton, FL, USA, 2018.
[22]. da Silva, A.J.; Ludermir, T.B.; de Oliveira, W.R. Quantum Perceptron over a Field and Neural Network Architecture Selection in a Quantum Computer. Neural Netw. 2016, 76, 55–64.
[23]. 23 17. Li, J.; Lin, S.; Yu, K.; Guo, G. Quantum K-Nearest Neighbor Classification Algorithm Based on Hamming Distance. Quantum Inf. Process. 2022, 21,
[24]. Lu, S.; Braunstein, S.L. Quantum Decision Tree Classifier. Quantum Inf. Process. 2014, 13, 757–770.
[25]. G.Viswanath & Dr.G.Swapna “Health Prediction Using Machine Learning with Drive HQ Cloud Security” in Frontiersin Health Informatics, 2024. Vol. 13, No. 8, pp. 2755-2761.
[26]. Saini, S.; Khosla, P.; Kaur, M.; Singh, G. Quantum Driven Machine Learning. Int. J. Theor. Phys. 2020, 59, 4013–4024.

[27]. Tang, E. A Quantum-Inspired Classical Algorithm for Recommendation Systems. In Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, ACM, Phoenix, AZ, USA, 23–26 June 2019; pp. 217–228.
[28]. Havlíˇcek, V.; Córcoles, A.D.; Temme, K.; Harrow, A.W.; Kandala, A.; Chow, J.M.; Gambetta, J.M. Supervised Learning with Quantum-Enhanced Feature Spaces. Nature 2019, 567, 209–212.
[29]. Shan, Z.; Guo, J.; Ding, X.; Zhou, X.; Wang, J.; Lian, H.; Gao, Y.; Zhao, B.; Xu, J. Demonstration of Breast Cancer Detection Using QSVM on IBM Quantum Processors. Res. Sq. 2022; preprint, in review.
[30]. Kumar, T.; Kumar, D.; Singh, G. Performance Analysis of Quantum Classifier on Benchmarking Datasets. IJEER 2022, 10, 375–380.
[31]. Kavitha, S.S.; Kaulgud, N. Quantum Machine Learning for Support Vector Machine Classification. Evol. Intel. 2022, 1–10.
[32]. Batra, K.; Zorn, K.M.; Foil, D.H.; Minerali, E.; Gawriljuk, V.O.; Lane, T.R.; Ekins, S. Quantum Machine Learning Algorithms for Drug Discovery Applications. J. Chem. Inf. Model 2021, 61, 2641–2647.
[33]. Suzuki, Y.; Yano, H.; Gao, Q.; Uno, S.; Tanaka, T.; Akiyama, M.; Yamamoto, N. Analysis and Synthesis of Feature Map for Kernel-Based Quantum Classifier. Quantum Mach. Intell. 2020, 2, 9.
[34]. Bai, Y.; Han, X.; Chen, T.; Yu, H. Quadratic Kernel-Free Least Squares Support Vector Machine for Target Diseases Classification. J. Comb. Optim. 2015, 30, 850–870
[35]. H.I. Garcıa-Hernandez, R. Torres-Ruiz, and G.H. Sun. “Image classification via quantum machine learning.” arXiv:2011.02831, 2020.
[36]. P. Rebentrost, M. Mohseni, and S. Lloyd. “Quantum support vector machine for big data classification.” Phys. Rev. Lett., vol. 113, no. 13, Art. no. 130503, doi: 10.1103/physrevlett.113.130503, 2014.
[37]. E. Farhi and H. Neven. “Classification with quantum neural networks on near term processors.” arXiv:1802.06002, 2018.
[38]. N. Meinhardt, B. Dekker, N. Neumann, and F. Phillipson. “Implementation of a Variational Quantum Circuit for Machine Learning with Compact Data Representation.” First International Symposium on Applied Artificial Intelligence, Munchen, 2019
[39]. E. Grant, M. Benedetti, S. Cao, A. Hallam, J. Lockhart, V. Stojevic, A.G. Green, and S. Severini. “Hierarchical quantum classifiers npj Quantum Inf.”, 2018.
[40]. H. Chen, L. Wossnig, S. Severini, H. Neven, and M. Mohseni. “Universal discriminative quantum neural networks.” arXiv:1805.08654, 2018.
[41]. A. Nakayama, K. Mitarai, L. Placidi, T. Sugimoto, and K. Fujii. “VQE-generated Quantum Circuit Dataset for Machine Learning.” arXiv:2302.09751, 2023.
[42]. G.Viswanath & G.Swapna, “Data Mining-Driven Multi-Feature Selection for Chronic Disease Forecasting”, in Journal of Neonatal Surgery,2025, Vol.14, No. 5s, pp. 108-124.
[43]. Rebentrost, P.; Mohseni, M.; Lloyd, S. Quantum support vector machine for big data classification. Phys. Rev. Lett. 2014, 113, 130503.
[44]. Zeguendry A, Jarir Z, Quafafou M. Quantum machine learning: A review and case studies. Entropy. 2023;25(2):287.
[45]. Mishra N, Kapil M, Rakesh H, et al. Quantum machine learning: A review and current status. Data Manage Anal Innov: Proc ICDMAI. 2020;2(2021):101-145
[46]. ZZFeatureMap. https://qiskit.org/documentation/stubs/qiskit.circuit.library.ZZFeatureMap.html
[47]. RealAmplitudes, https://docs.quantum.ibm.com/api/qiskit/0.24/qiskit.circuit.library.RealAmplitudes
[48]. EfficientSU2, https://docs.quantum.ibm.com/api/qiskit/qiskit.circuit.library.EfficientSU2
[49]. Breast-cancer Data, https://www.kaggle.com/uciml/ breast-cancer-wisconsin-data
[50]. Tychola, K.A.; Kalampokas, T.; Papakostas, G.A. Quantum Machine Learning—An Overview. Electronics 2023, 12, 2379. https:// doi.org/10.3390/electronics12112379.
[51]. S. Yasmin and G. Murali, "A new framework for analyzing surveillance of quantum cryptographic protocols using genetic algorithm," 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS), Chennai, India, 2017, pp. 3599-3603, doi: 10.1109/ICECDS.2017.8390133.
[52]. S. Sreelatha and G. Murali, "Error correction and detection techniques in quantum cryptography protocol," 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS), Chennai, India, 2017, pp. 3584-3588, doi: 10.1109/ICECDS.2017.8390130.
[53]. P. S. Lakshmi and G. Murali, "Comparison of classical and quantum cryptography using QKD simulator," 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS), Chennai, India, 2017, pp. 3543-3547, doi: 10.1109/ICECDS.2017.8390120.